Python for Financial Research

Dr. Vincent Grégoire
Department of Finance

March 2017

THE UNIVERSITY OF MELBOURNE

Why learn to code?

- It's fun
- Saves you time
- Produce better research
- Looks good on your CV
- Likely to help your career down the road

Why learn to code?

Automation is already here

"At its height back in 2000, the U.S. cash equities trading desk at Goldman Sachs's New York headquarters employed 600 traders, buying and selling stock on the orders of the investment bank's large clients. Today there are just two equity traders left.

Automated trading programs have taken over the rest of the work, supported by 200 computer engineers."

–MIT Technology Review, February 2017

Why Python?

- General purpose programming language
  - Websites (i.e. Instagram)
  - Games, full software
- Free, open source software
- Very large community
- Tons of online resources and textbooks
- Integrates nicely with R, C/C++ and other languages
- More general than domain-specific software:
  - Matlab
  - R
  - SAS
  - Stata
  - EViews
  - etc...

Why Python?

- What is it very good for?
  - Data manipulation
  - Visualization
  - Web scraping
  - Text analysis
  - Basic to somewhat advanced statistics and econometrics
  - Linear algebra

- What is it less good for
  - Multithreaded, very high-speed concurrent applications
  - Advanced statistical analysis

Main scientific Python libraries

NumPy

The main package for numerical analysis in Python. Includes functions to deal with arrays of data, linear algebra, random number generation, and much more.

Main scientific Python libraries

NumPy

The main package for numerical analysis in Python. Includes functions to deal with arrays of data, linear algebra, random number generation, and much more.

pandas

Package for panel data analysis, built on top of NumPy. Great for importing/exporting, merging, cleaning and analysing data. Written by Wes McKinney while working as a quant at AQR, so very good for dealing with financial data.

Main scientific Python libraries

NumPy

The main package for numerical analysis in Python. Includes functions to deal with arrays of data, linear algebra, random number generation, and much more.

pandas

Package for panel data analysis, built on top of NumPy. Great for importing/exporting, merging, cleaning and analysing data. Written by Wes McKinney while working as a quant at AQR, so very good for dealing with financial data.

rpy2

Use R functions with pandas panels.

Main scientific Python libraries

SciPy

A collection of packages that add a lot of nice features such as more advanced linear algebra (above NumPy), numerical integration and optimization.

Main scientific Python libraries

SciPy

A collection of packages that add a lot of nice features such as more advanced linear algebra (above NumPy), numerical integration and optimization.

statsmodels

Statistical analysis, including OLS regressions with support for fixed effects and clustering.

Main scientific Python libraries

SciPy

A collection of packages that add a lot of nice features such as more advanced linear algebra (above NumPy), numerical integration and optimization.

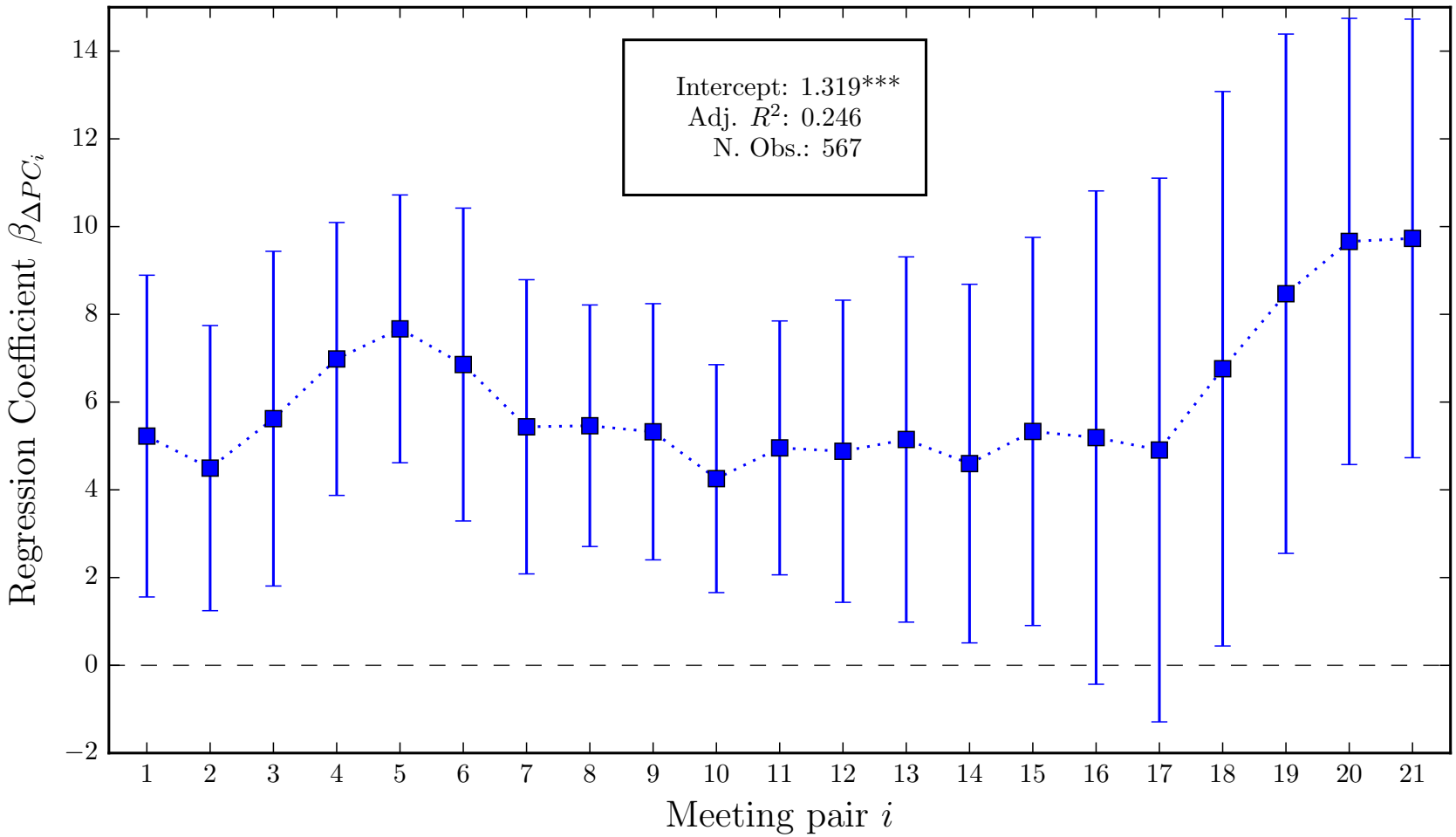
statsmodels

Statistical analysis, including OLS regressions with support for fixed effects and clustering.

matplotlib

Main package for visualization in Python. Can produce many type of graphs, highly customizable.

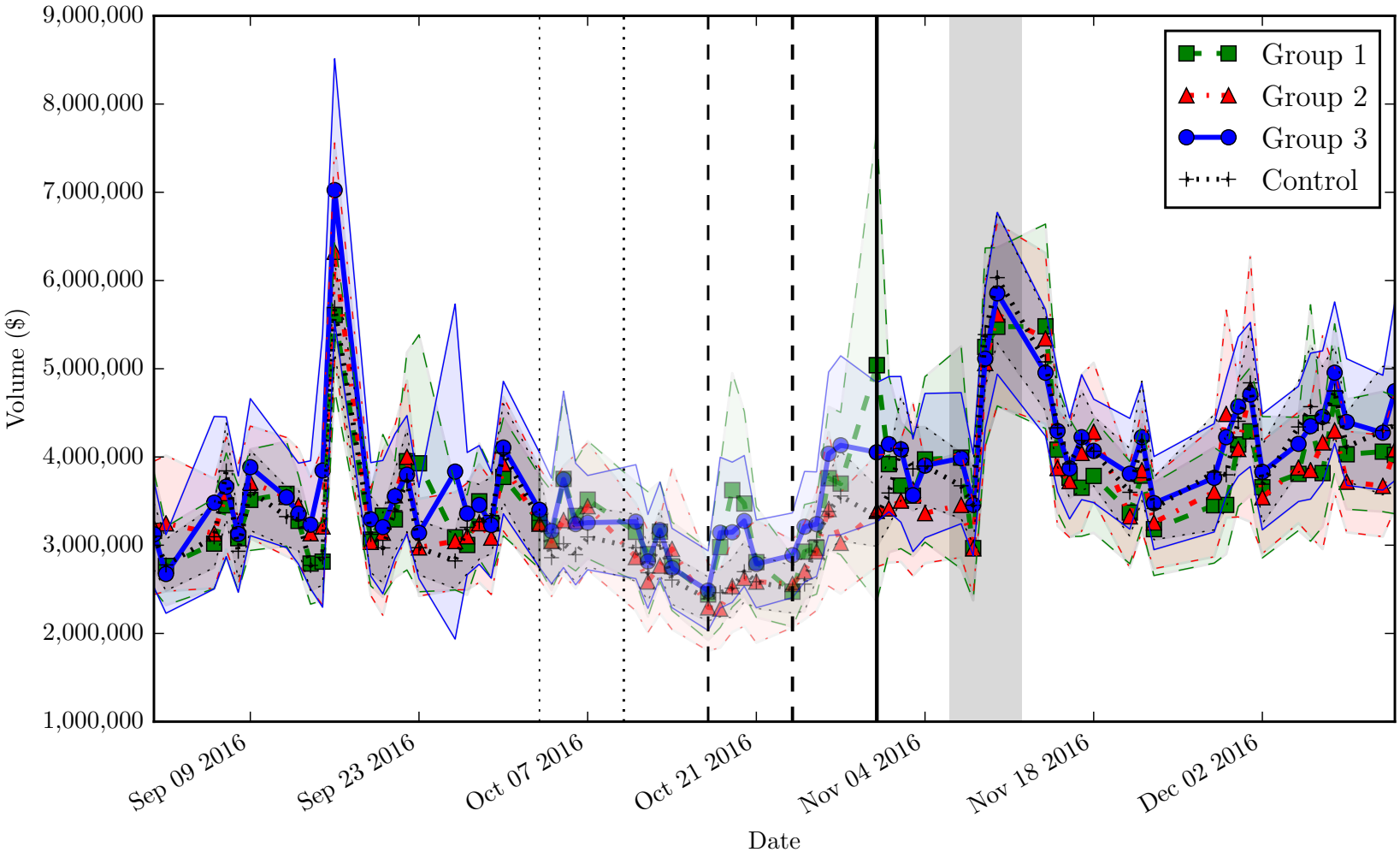
Sample Matplotlib plots



Source: Boguth, Gregoire and Martineau, *Shaping Expectations and Coordinating Attention: The Unintended Consequences of FOMC Press Conferences*, 2016.

Available at: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2698477`

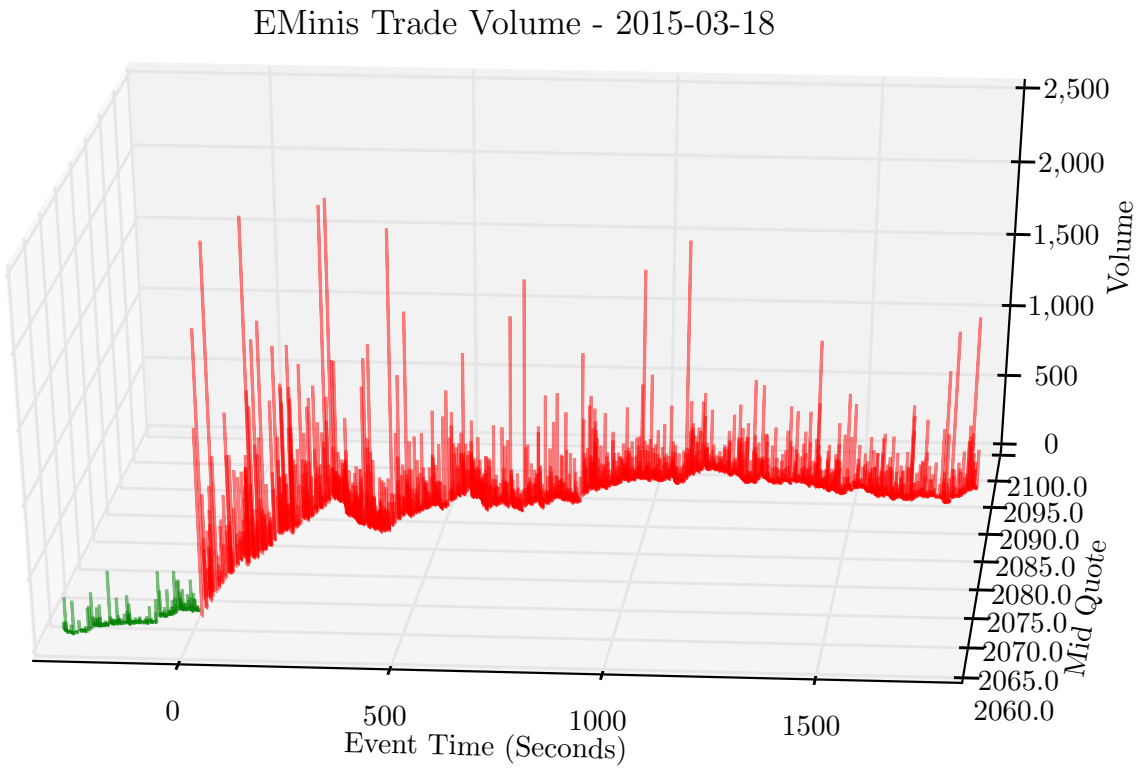
Sample Matplotlib plots



Source: Comerton-Forde, Gregoire and Zhong, *Inverted Fee Venues and Market Quality*, 2017.

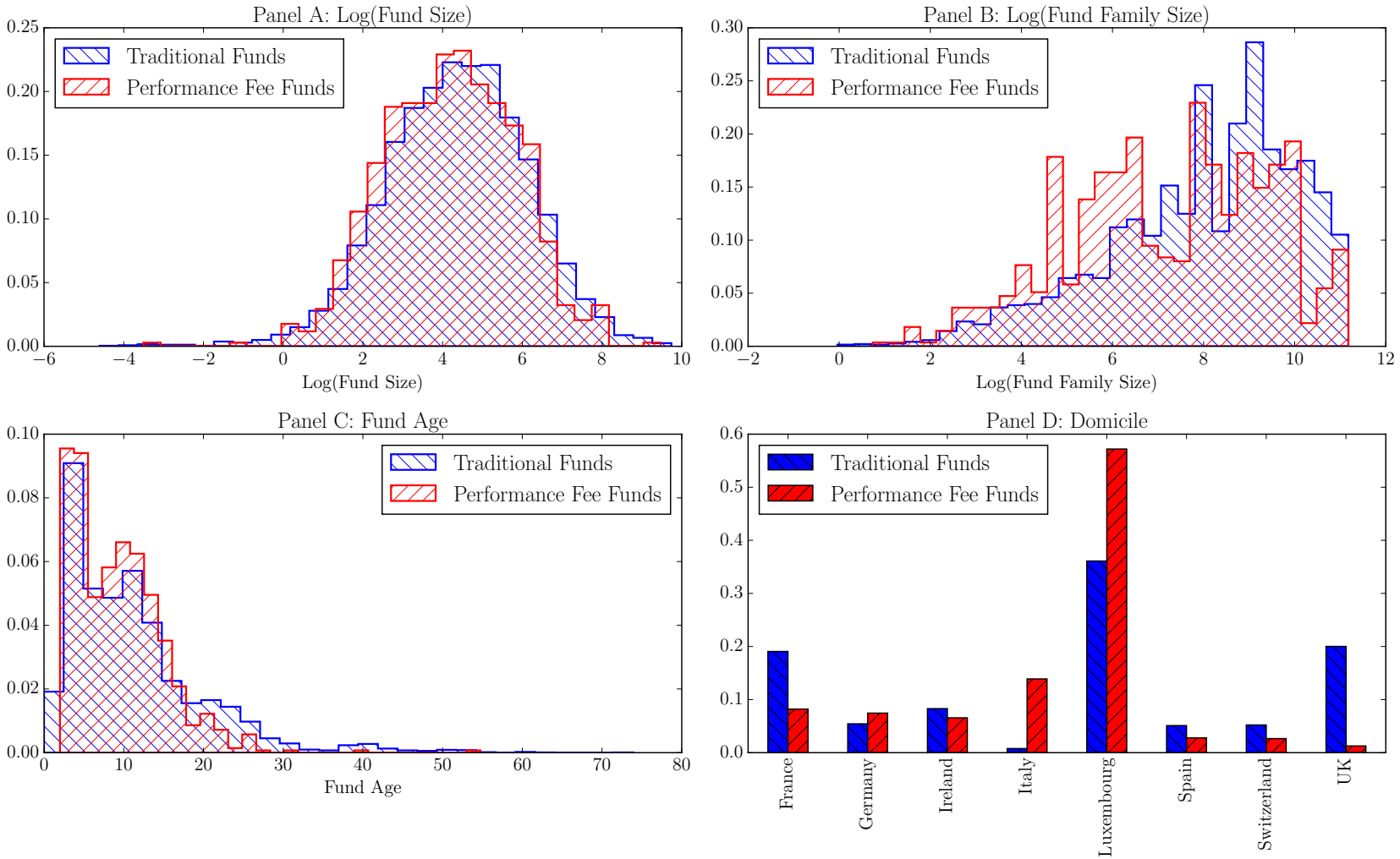
Available at: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=`

Sample Matplotlib plots



Source: Boguth, Gregoire and Martineau, *Price Formation around FOMC Announcements*, 2017. (Work in progress)

Sample Matplotlib plots



Source: Gregoire and Sotes-Paladino, *Double Bonus? Implicit incentives in Mutual Funds with Explicit Performance Fee*, 2017. (Work in progress)

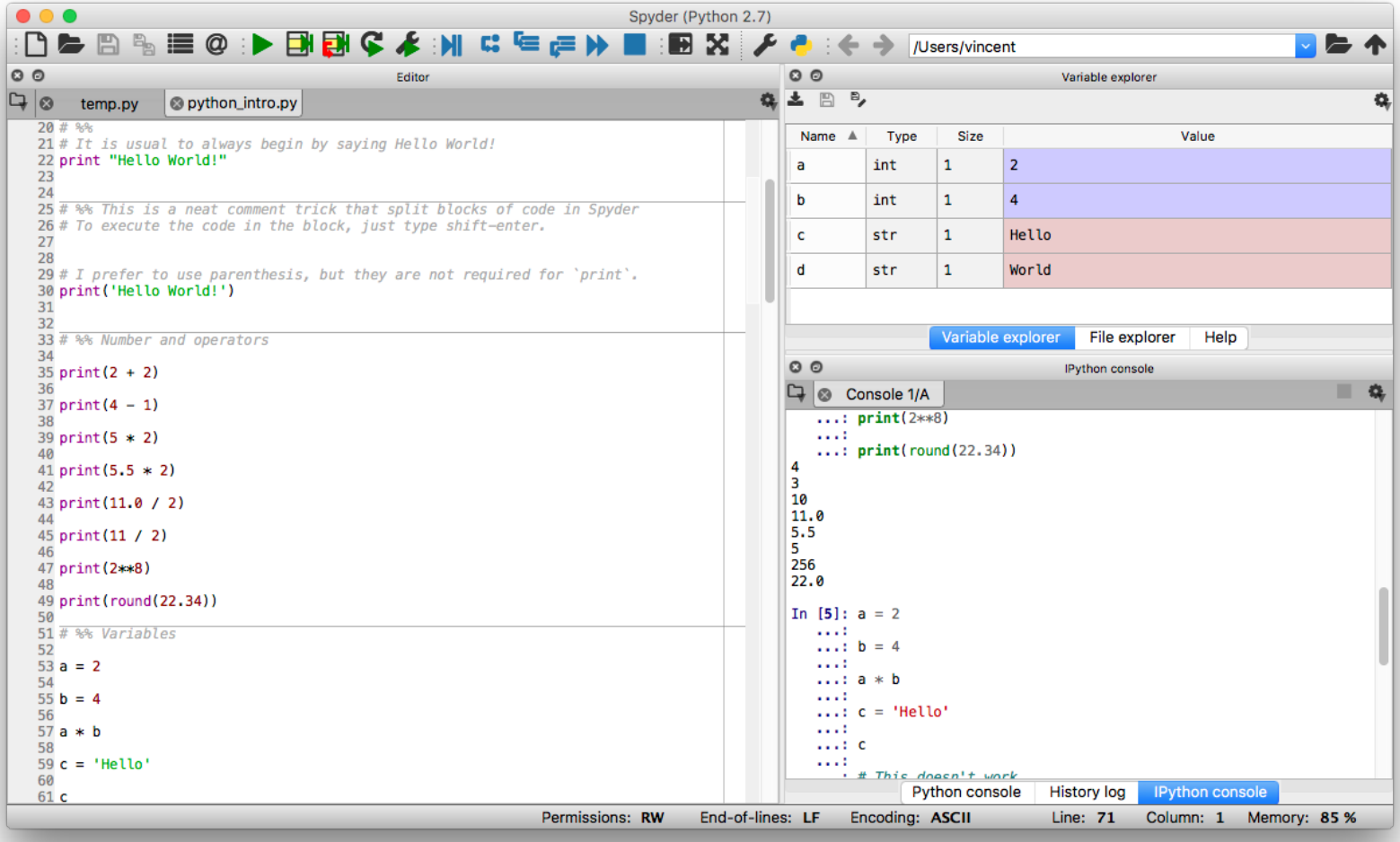
Additional useful libraries

- plotly, Bokeh: other plotting libraries
- SymPy: symbolic algebra (think Mathematica or Maple)
- NetworkX: network analysis
- scikit-learn: machine learning
- NLTK, textblob: textual analysis
- Scrapy, Beautiful Soup: webscraping and information extraction
- Plus many more! Just search for what you want to do, chances are there is a Python package for it.

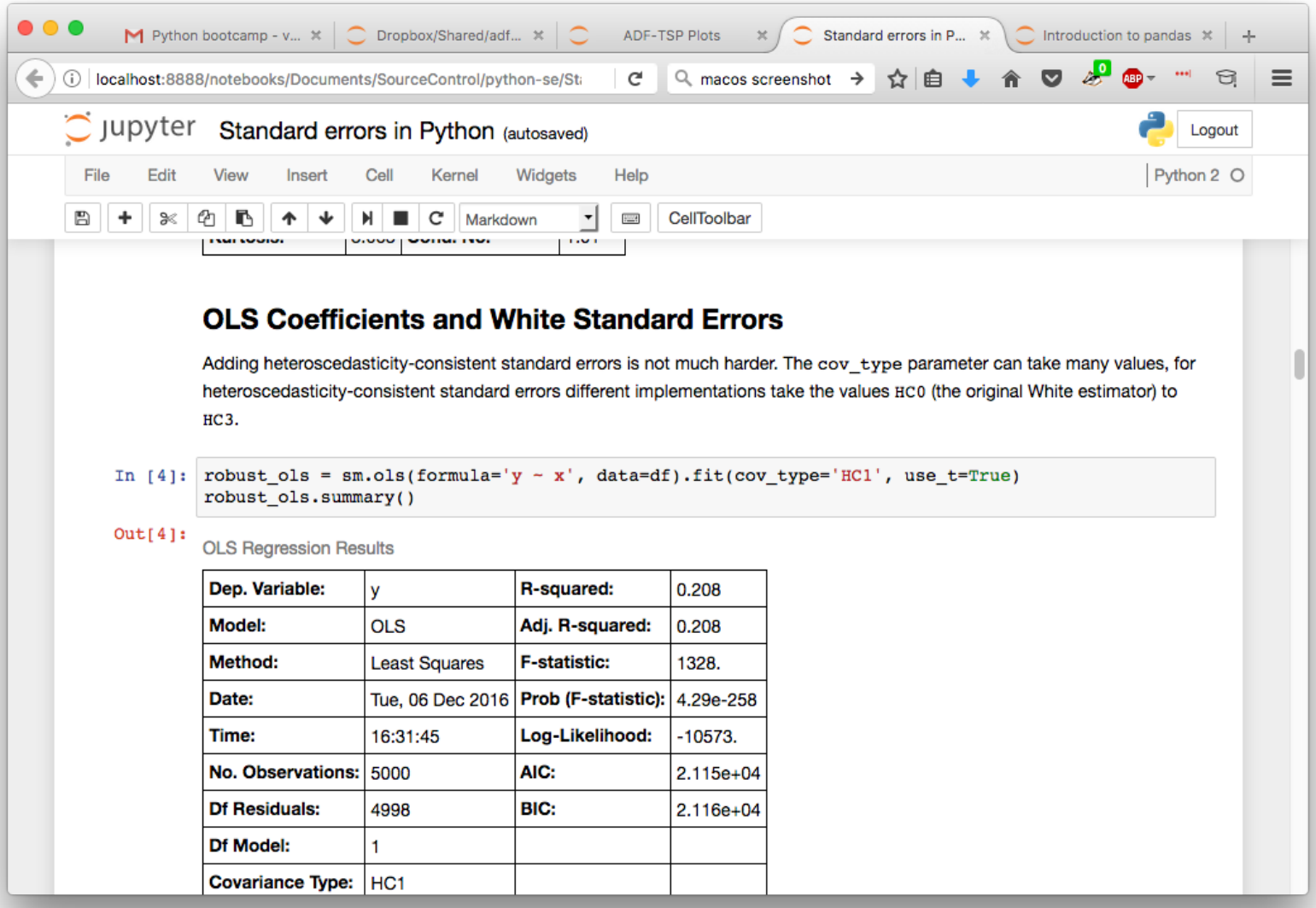
Anaconda

- The easiest way to install Python with scientific packages is with the Anaconda distribution by Continuum Analytics:
  - Mac OS, Windows, Linux
  - Python 2 or 3 (we use 2.7 in this tutorial)
  - `https://www.continuum.io/downloads`
- Includes hundreds of packages by default.
- Easily manage "environments" (sets of packages) for each project.
- Includes Spyder and Jupyter, and R as an option.

Spyder


Spyder (Python 2.7)
/Users/vincent
Editor
temp.py
python_intro.py
%%
It is usual to always begin by saying Hello World!
print "Hello World!"
# %% This is a neat comment trick that split blocks of code in Spyder
# To execute the code in the block, just type shift-enter.
# I prefer to use parenthesis, but they are not required for `print`.
print('Hello World!')
# %% Number and operators
print(2 + 2)
print(4 - 1)
print(5 * 2)
print(5.5 * 2)
print(11.0 / 2)
print(11 / 2)
print(2**8)
print(round(22.34))
# %% Variables
a = 2
b = 4
a * b
c = 'Hello'
c
Variable explorer
Name
Type
Size
Value
a
int
1
2
b
int
1
4
c
str
1
Hello
d
str
1
World
Variable explorer
File explorer
Help
IPython console
Console 1/A
...: print(2**8)
...: print(round(22.34))
4
3
10
11.0
5.5
5
256
22.0
In [5]: a = 2
...: b = 4
...: a * b
...: c = 'Hello'
...: c
Python console
History log
IPython console
Permissions: RW
End-of-lines: LF
Encoding: ASCII
Line: 71
Column: 1
Memory: 85 %

Jupyter (IPython notebook)

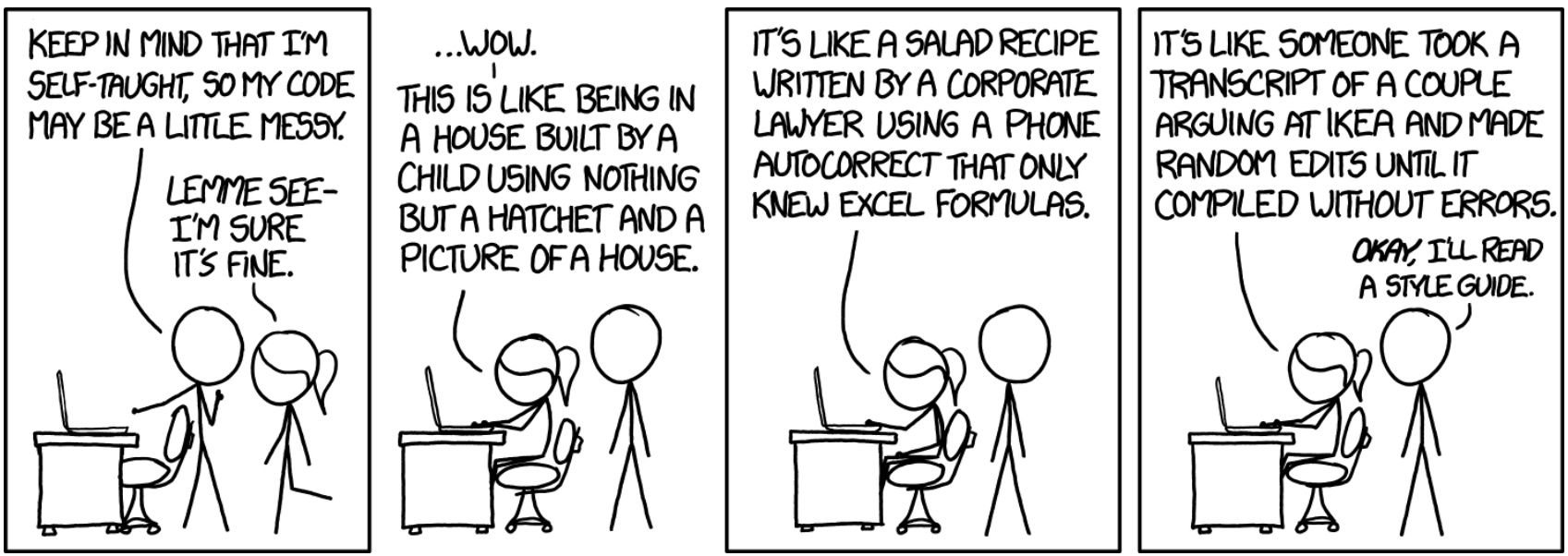


Keep you code clean

- Write code that is easy to read, and document it well.
- Will save you time in the long run, easier to spot errors.
- Also easier to get feedback and help!
- Python style guide is called PEP8.
  - `https://www.python.org/dev/peps/pep-0008/`

Keep you code clean

- Write code that is easy to read, and document it well.
- Will save you time in the long run, easier to spot errors.
- Also easier to get feedback and help!
- Python style guide is called PEP8.
  - `https://www.python.org/dev/peps/pep-0008/`



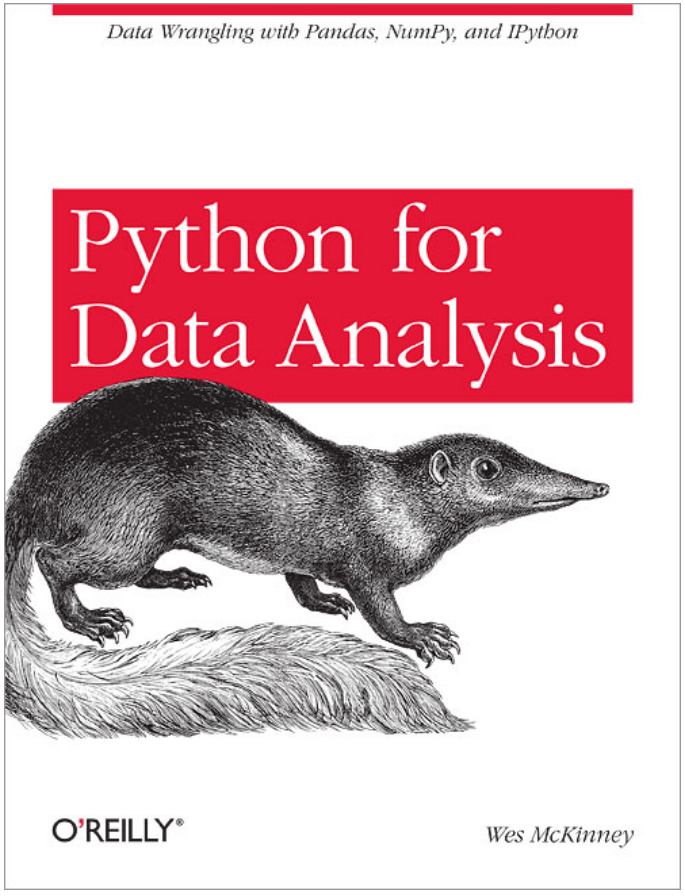
Credit: `http://xkcd.com`

Online resources

- Slides and code from this bootcamp, and more on my website
  - `http://www.vincentgregoire.com/python-bootcamp/`
- Python documentation
  - `https://docs.python.org/2/reference/index.html`
  - `https://docs.python.org/2/library/index.html`
- LearnPython.org
  - `http://www.learnpython.org/`
- Package documentation
  - `https://docs.scipy.org/doc/numpy/index.html`
  - `http://pandas.pydata.org/pandas-docs/stable/`
  - `http://matplotlib.org/contents.html`
- Stack Overflow
  - `http://stackoverflow.com/`
- Google / DuckDuckGo / Bing (?)

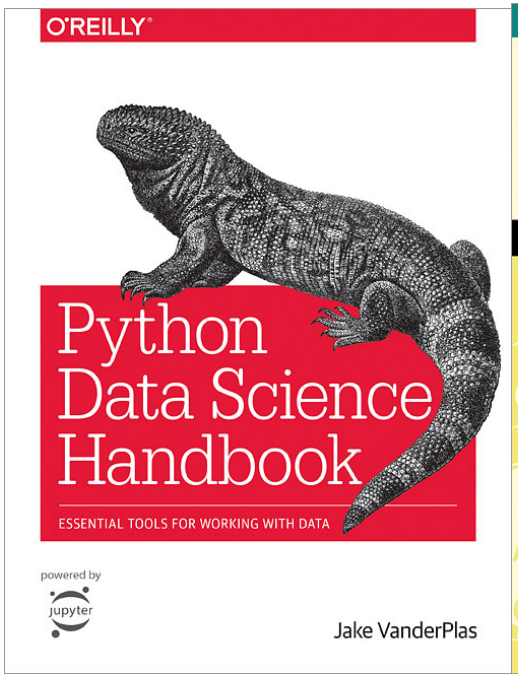
Books

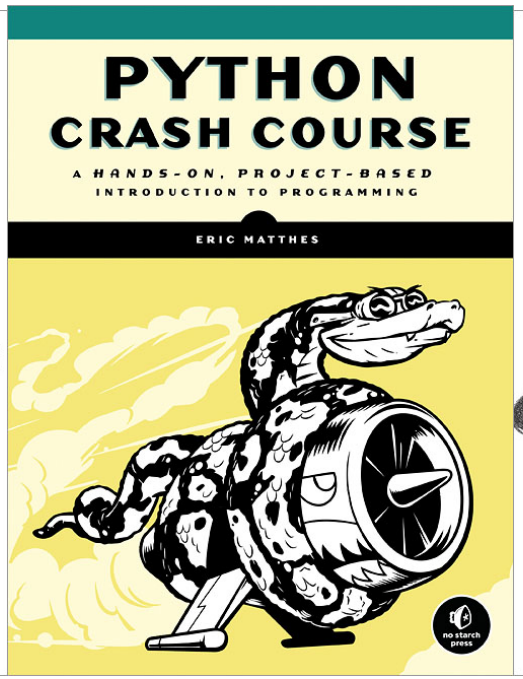
If there's a book you need to get, it's this one.

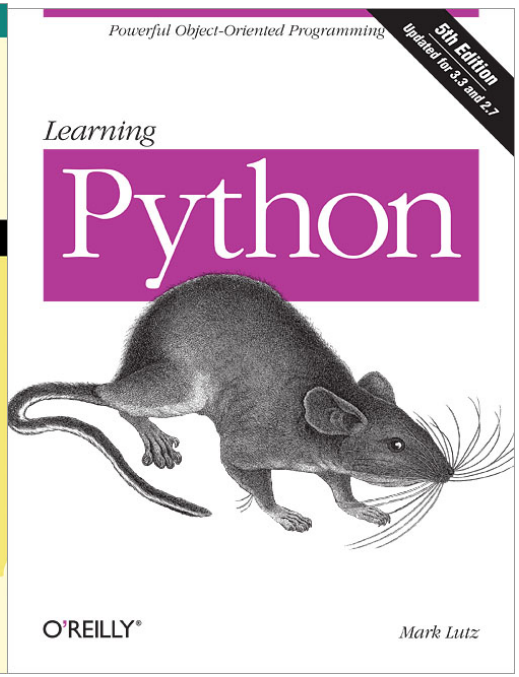


Note: First edition is from 2012, new one coming out in 2017. But it's cheap, so don't wait, get it now.

Books


O'REILLY
Python
Data Science
Handbook
ESSENTIAL TOOLS FOR WORKING WITH DATA
powered by
jupyter
Jake VanderPlas


PYTHON
CRASH COURSE
A HANDS-ON, PROJECT-BASED
INTRODUCTION TO PROGRAMMING
ERIC MATTHES
no starch press


Powerful Object-Oriented Programming
5th Edition
Updated for 3.3 and 2.7
Learning
Python
O'REILLY
Mark Lutz

Other (free, open source) tools for research

- Markdown: Easy markup language that can be used in Jupyter notebooks. Atom is a nice editor that supports Markdown.
  - `http://daringfireball.net/projects/markdown/`
  - `https://atom.io/`

- LaTex: Typesetting system to produce nice report papers (and presentations such as this one). You can export pandas datasets directly to LaTeX tables, and import in LaTeX the PDF figures created in Matplotlib.
  - `https://www.tug.org/begin.html`

- R: Statistical computing, more advanced functions than currently in Python. You can install R through Anaconda, and call R code from within Python.
  - `https://www.r-project.org/`

- git: Version control, think "track changes" for your code.
  - `https://git-scm.com/doc`

Outline

Four blocks of three hours:

1. Introduction to Python programming

We will discuss what is Python and you will learn the basic structure of the language. You will also learn our way around the programming environment, including the two main editors for scientific Python, Spyder and Jupyter.

Outline

Four blocks of three hours:

1. Introduction to Python programming

We will discuss what is Python and you will learn the basic structure of the language. You will also learn our way around the programming environment, including the two main editors for scientific Python, Spyder and Jupyter.

2. Introduction to data analysis using pandas, matplotlib

You will learn how to import, export and transform data using pandas, the panel data package for Python. You will also learn how to explore the data by generating summary statistics and plotting graphs using matplotlib.

Outline

3. More data analysis using pandas and statsmodels

You will learn more advanced features of Python and pandas, including dealing with timestamps and estimating measures from daily and intraday data. You will also learn how to estimate OLS and panel regressions using statsmodels.

Outline

3. More data analysis using pandas and statsmodels

You will learn more advanced features of Python and pandas, including dealing with timestamps and estimating measures from daily and intraday data. You will also learn how to estimate OLS and panel regressions using statsmodels.

4. Other topics

In this block, you will be introduced briefly to other Python packages that can be helpful for research. The list of topics is not yet finalized, but will likely include text analysis, web scraping, network analysis and symbolic algebra.

Let's get started!